REMARKS

Claims 1 – 19 are pending in the present application.

Claims 1-6 and 13-17 stand rejected under 35 U.S.C. § 112, first paragraph. Applicant traverses this rejection. In particular, the examiner states this rejection is asserted because each of claims 1 and 13 recite the term "executable" which was not described in the specification. In paragraph 5 of the Office Action, the examiner states the following:

> "As pointed out in the 35 U.S.C. 112 rejection above, the term "executable" is not described in the specification. In the specification, it is noted that the list container object may only "instantiate" a number of item renderer objects corresponding to list item data objects as described on page 4, lines 11-12 of the specification. Nowhere can the term "executable" be found under the same context as claimed. The terms "executable" and "instantiate" are very different in meaning. By definition, "executable" means pertaining to a program file that can be run, "instantiate" means to create/generate an instance of. Thus, the claims are interpreted by the Examiner as "instantiate" in light of the specification."

Applicant makes the following observations concerning the examiner's statement above.

First, the examiner is correct in stating that the term "executable" is not described in the specification. However, Applicant submits such a description is not required. The term executable is so widely known in the art that a description or definition is not necessary, unless something other than a common meaning is intended.

Second, the examiner states that in the specification, a list container object may only "instantiate" a number of item renderer objects corresponding to list item data objects. However, such an assertion simply ignores the remainder of the description. List container objects are describes as being able to perform a wide variety of functions. For example, the following list provides only some of the examples from the description of functions which may be performed by list container objects:

> "The list container object may provide an API enabling the client program to then add "list item data objects" to the list container object." (page 4, lines 6-7).
>
> "The list container object may maintain these list item data objects in any of various ways as appropriate to a particular implementation or programming environment." (page 4, lines 7-9).
>
> "The list container object may instantiate a fixed number of "item renderer objects", which are responsible for appropriately displaying the list item data objects." (page 4, lines 11-12).
>
> "The list container object interfaces with the set of item renderer objects, in order to manage the display of the list." (page 4, lines 13-14).
>
> "The list container object may keep track of the set of list item data objects being displayed at any given time." (page 4, lines 21-22).
>
> "As a user interacts with the list, e.g., by scrolling up or down, the list container object may receive user interface events indicating the user's action, may determine the new start index for items to display, and may instruct each item renderer object to redisplay." (page 4, lines 23-25).
>
> "[T]he list container object may provide a method with a signature such as: public synchronized void addItem (Object item)." (page 10, lines 21-23).
>
> "For example, the list container object may include a SetRendererClass() method callable by client programs." (page 13, lines 9-10).
>
> "In response to the list container object setting the data for the list item renderer objects, each list item renderer object may interface with its associated user interface component to display the data appropriately." (page 13, line 28 – page 14, line 2).
>
> "[T]he list container object to cause the displayed list to scroll, etc." (page 15, lines 5-6).

> "In step 206, the list container object may also set the corresponding data object for each list item renderer object." (page 16, lines 17-18).

> "In step 210, the list container object begins processing events, such as user interface events." (page 17, lines 1-2).

Accordingly, list container objects are capable of performing a wide variety of functions. As may be appreciated, the list container object may generally include code which is executable to perform these functions. It is well known in the art that objects include code which is executable. Therefore, a specific definition or description of "executable" is not believed necessary and the recitation "wherein the list container object is executable to specify a corresponding list item data object for each of a plurality of list item renderer objects" is believed to be in compliance with 35 U.S.C. § 112.

Third, the examiner states the following:

> "The terms "executable" and "instantiate" are very different in meaning. By definition, "executable" means pertaining to a program file that can be run, "instantiate" means to create/generate an instance of."

Applicant would suggest that "executable" does not necessarily mean a "program file" that can be run. Rather, program instructions, or even a single program instruction, are executable. In addition, executable instructions need not be stored in a "file", but may simply be memory resident. In any event, within the context of the presently claimed invention, those skilled in the art will readily understand that an object may include code which is executable. The description makes several references to programming code and programming languages in relation to the described invention. It is further noted that the examiner admits the list container object may instantiate item renderer objects. As may be appreciated, the act of instantiation itself generally entails the execution of executable code.

In view of the discussion above, Applicant submits the term executable is both acceptable and appropriate, and the 35 U.S.C. § 112 rejection should be withdrawn.

Applicant notes that Claims 1-19 stand rejected under 35 U.S.C. 102(e) as being anticipated by Pogue (PalmPilot: The Ultimate Guide, hereinafter "Pogue"). In view of the above discussion, Applicant submits the 35 U.S.C. § 112 rejection is inappropriate and Applicant's previous arguments against the rejections in view of Pogue remain and are reasserted without being repeated herein. Accordingly, Applicant submits all rejections are overcome.

Finally, the examiner states "[t]hus, the claims are interpreted by the Examiner as "instantiate" in light of the specification." Applicant has already discussed why the term executable is appropriate and the adopted interpretation of "instantiate" would therefore be inappropriate. Nevertheless, Applicant notes that even if the interpretation suggested by the examiner were adopted, such an interpretation would not support the 35 U.S.C. § 102(e) rejections made by the examiner. As noted above, in paragraph 5 of the present Office Action, the examiner rejects the term "executable" and substitutes the term "instantiate". As the examiner has equated the recited "list container object" with a directory, the examiner is apparently suggesting that the cited art then teaches a directory which "instantiates" data objects. However, such a teaching is wholly absent from the cited art. Therefore, even assuming the examiner's own interpretation, the 35 U.S.C. § 102 rejection is inappropriate and should be withdrawn.

Should the examiner wish to discuss this further, the below signed representative would appreciate a phone call at (512) 853-8866 in order to facilitate a resolution.

CONCLUSION

In light of the foregoing remarks, Applicant respectfully submits the application is now in condition for allowance, and an early notice to that effect is requested.

No fees are believed necessary; however, the Commissioner is authorized to charge any fees which may be required, or credit any overpayment, to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account No. 50-1505\5181-53800\RDR.

Respectfully submitted,

[illegible]

Rory D. Rankin
Reg. No. 47,884
ATTORNEY FOR APPLICANT

Meyertons, Hood, Kivlin,
Kowert & Goetzel, P.C.
P.O. Box 398
Austin, Texas 78767-0398
Phone: (512) 853-8800

Date: Nov. 30, 2004